# Colour-Range Histogram Technique for Automatic Image Source Detection

Nancy C. Woods and Abiodun B. C. Robert
Department of Computer Science, University of Ibadan, Ibadan, Nigeria
E-mail: Chyn.woods@gmail.com and abc.robert@live.com

**Keywords:** natural images, computer generated images, colour histogram

**Received:** November 28, 2018

*Computer generated images are visually becoming increasingly genuine, due to advances in technology as well as good graphic applications. Consequently, making distinction between computer generated images and natural images is no longer a simple task. Manual identification of computer generated images have failed to resolve the problems associated with legal issues on exact qualification of images. In this work, a colour range histogram was developed to categorise colours in computer generated images and natural images from a point of reference. Four groups were selected, using the algorithm, consisting of exact Red-Green-Blue (RGB) code (group 1), colour code within a range of 10 (group 2), colour code within a range of 20 (group 3) and colour code within a range of 30 (group 4) from the point of reference. An optimised equation for the four Colour Code Groups (CCG) was developed. The computer generated images categorised an average of 69.8%, 92.9%, 96.9% and 98.6%, of any colour code for groups 1, 2, 3 and 4, respectively. The categorised colours for natural images were 31.1%, 82.6%, 90.8% and 95.0% for groups 1, 2, 3 and 4, respectively. The results showed that natural images contain a wide range of RGB colours which makes them different. Consequently, the disparity in the percentage of colours categorised can be used to differentiate computer generated images from natural images.*

*Povzetek: Razvit je sistem za razločevanje naravnih od računalniško generiranih umetnih slik na osnovi barvnega histograma.*

## 1 Introduction

Digital images have become a commonplace in the lives of individuals nowadays, because of the ease of acquisition using mobile phones and other electronic devices. A digital image can be described as a rectangular two-dimensional array of pixels, where each pixel (usually a square) represents the image colour at that position and where the dimensions represent the width and height of the image as it is displayed [1]. With advances in technology and the proliferation of imaging software, digital images are now classified as either *computer generated* or *natural*. With image processing techniques, it is becoming increasingly easy to produce computer-generated images (CGI) that are so realistic with commercially available software packages [2] and these CGI are presently called Photorealistic images. How can one tell if a digital image is natural or computer generated? Usually, a photograph provides an effective and natural communication medium for humans. This is because people do not really need any special training to comprehend the content of an image and they used to believe that photographs represent the truth [3]. Unfortunately, this truth no longer holds with digital images because it is easy to manipulate them [4]. Therefore, being able to verify the credibility of digital images and perform image forensics can protect the truthfulness of digital images. It can be cumbersome and difficult for the human eye to tell the difference between the two types of images [5]. This is what the research carried out under the field of digital image forensics, among other things tries to answer. Digital image forensics is the area of image processing with the main function of assessing the authenticity and the origin of images and is divided into *active* forensics and *passive* forensics, which are further sub divided [6], [3], [4]. Figure 1 shows the classification of digital image forensics. In active forensics, additional information needs to be inserted into the host or source of the image in advance. This requires that the acquisition device should have the corresponding functionality to hold such information, some of which include digital signature [7] or digital watermarking [8]. Passive forensics technology is more practical and attempts to identify the authenticity or source of an image, based only on the characteristics of the image itself without embedded additional information [6]. Passive forensics occurs after the image has been captured and stored. Depending on its applications in different research fields, passive forensics can be broadly classified into ***tampering detection*** [9], [10], [11], ***Steganalysis*** [12] and ***source identification*** [13], [6], which is the art and science of differentiating computer generated images from natural images (NI).

The aim of this work therefore is to develop a model for colour range histogram towards discovering features that differentiate CGI from NI.

## 2 Literature review

Several research has been carried out in a bid to differentiate computer generated images from natural images using several approaches and features. One of the first approaches offered to differentiate NI from CGI was proposed by [14]. In their statistical approach, the first and higher-order statistics of wavelet transform coefficients are extracted from both CGI and NI to capture their statistical regularities. Another work by [15] proposed an approach using differences in image texture by considering the physical / visual properties of these images. They took into account the differences in surface and object models as well as the differences in the acquisition processes between the CGI and NI, and extracted 192 geometry features by analysing the differences existing between the physical generative process of computer graphics and photographs. Their approach extracted a lot of features in a bid to find the difference between CGI and NI.

A total of 216 features, based on the RGB colour information of CGI and NI were considered and extracted by [12] in their work. As such, their method employed image decomposition based on separable quadrature mirror filters (QMFs) to capture regularities inherent to photographic images [12]. An approach presented by [16] discriminates CGI from NI based on the lack of artifacts due to the use of a digital camera as an acquisition device for NI [16]. Their technique is based on the fact that image acquisition in a digital camera is fundamentally different from the generative algorithms deployed by computer generated imagery. This difference is captured in terms of the properties of the residual image (pattern noise in case of digital camera images) extracted by a wavelet based de-noising filter. An approach proposed by [17] used features that are based on the differences in the acquisition process of images. First they tried to detect the presence of the colour filter array demosaicking from a given image because most consumer cameras use colour filter array which requires the involvement of a demosaicking operation in generating the RGB colour values. The approach by [17] specifically searched for traces of demosaicking and chromatic aberration which were used to differentiate CGI from NI.

Another technique based on the differences in the acquisition process of images was proposed by [18]. The starting point of their research is that the different formation processes, leave distinct intrinsic traces on digital images. In their algorithm, spectral correlations between colour components are exploited efficiently by discrete wavelet transform, block partitioning and normalized cross correlation, and three statistical features are derived to capture the inherent differences between CGI and NI. [6] combined statistical, visual and physical features of digital images to propose features that can differentiate CGI from NI. Their approach amongst other features, extracted the mean and median of the histograms of grayscale image in the spatial and wavelet domain as statistical features. Secondly, the fractal dimensions of grayscale image and wavelet sub-bands were extracted as visual features. And finally, the physical features are calculated from the enhanced photo response non-uniformity noise. Thereafter, a support vector machine (SVM) classifier was used in the classification process.

More recently, the researchers in [19] comparing CGI with NI, extracted and used 9 dimensions of texture features. They argued that NI have higher self-similar and have more delicate and complex texture. The work by [5] extracted textural descriptors from images using binary statistical image features and also used SVM as the classifier. According to them, the textural features are different for CGI and NI as their approach was based on learning of natural image statistic filters and further using that to differentiate the two images.

In this research work, a model is proposed where colour and statistical features are extracted and combined in identifying features that differentiate CGI from NI.

## 3 Methodology

The proposed model is termed ***Colour Range Histogram*** (CRH). The CRH works by first randomly selecting a pixel $A_{x,y}$ in an image as a point of reference. Next, the

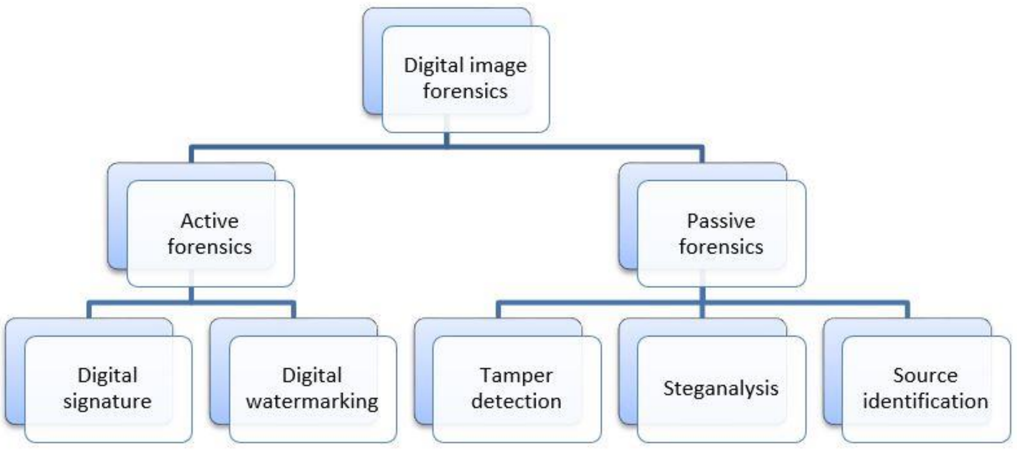

Figure 1: Classification of Digital Image Forensics.

CRH algorithm fetches the RGB colour code for $A_{x,y}$, which is an integer value, due to the programming language used. Then the algorithm checks through all other pixels in the rest of the image to highlight all pixels that have the same RGB colour code as pixel $A_{x,y}$. These pixels were classified as group 1 pixels. The complete steps used in CRH are further elucidated in the pseudo codes in Listing 1. In general, the following steps are proposed:

1. For any image (A), with dimension $(w \times h)$ where *h* represents the height and *w* represents the width of the image, select any pair of coordinates (*x*, *y*) that represent a pixel position such that $0 \geq x \leq w$ and $0 \geq y \leq h$.
2. Given $A_{x,y}$ fetch $RGB\ (A_{x,y})$ where $A_{x,y}$.represents a pixel in A at the pixel position (*x*, *y*)
3. Scan through all other pixels in (A), from $A_{0,0}$. to $A_{w,h}$.
    3.1 Fetch $RGB\ (A_{s,t})$
    3.2 If $RGB(A_{s,t})$ = $RGB(A_{x,y})$, retain its colour
    else
    change the colour of $A_{s,t}$ to white
    3.3 Fetch next pixel
4. Save the resultant image (A_m).

The algorithm above was used to "highlight" group 1 pixels, which are pixels with the exact RGB code as $A_{x,y}$. The algorithm was further extended to highlight seven more groups of pixels (Listing 2). These are pixels in the image that have RGB colour codes within certain ranges from $RGB\ (A_{x,y})$. These groups were:

| | |
|---|---|
| Group 2: | where $RGB(A_{s,t})$ is within ±10 from $RGB(A_{x,y})$ |
| Group 3: | where $RGB(A_{s,t})$ is within ±20 from $RGB(A_{x,y})$ |
| Group 4: | where $RGB(A_{s,t})$ is within ±30 from $RGB(A_{x,y})$ |
| Group 5: | where $RGB(A_{s,t})$ is within ±40 from $RGB(A_{x,y})$ |
| Group 6: | where $RGB(A_{s,t})$ is within ±50 from $RGB(A_{x,y})$ |
| Group 7: | where $RGB(A_{s,t})$ is within ±60 from $RGB(A_{x,y})$ |
| Group 8: | where $RGB(A_{s,t})$ is within ±70 from $RGB(A_{x,y})$ |

By 'highlight' we mean that their original colour is retained, while the colour of the rest of the image was changed to white. However, if the randomly selected pixel was white in colour then the colour of the rest of the image was changed to Black as can be seen from figure 2c. This highlight was to enable us have a visual clue of the resultant image.

In addition to saving the image file for a visual presentation of CRH, eight more features which represent the total number of pixels projected in each group was captured. The image dataset used for this work was obtained from the Internet as well as personal picture collections. A total of 1,620 images were obtained which contained 851 CGIs and 769 NIs.

```
Load imageA = ImageIO.read(new File(path))
    imageA_width = image.getWidth();
    imageA_height = image.getHeight();
    Pick $Pixel_{x,y}$ where $0 \geq x \leq$ imageA_width and $0 \geq y \leq$ imageA_height;
    Pcolour = imageA.getRGB(x,y);
  for (int w = 0; w < imageA_width(); w++)
  for (int h = 0; h < image_height(); h++){
    Pixelcolour = image.getRGB(w,h);
      if (Pcolour = pixelcolour)
                  retain pixel colour
      else
                  change pixel colour to white;
  }
        Save image;
```

Listing 1: Algorithm for exact colour highlight.

```
Load image = ImageIO.read(new File(path))
    width = image.getWidth();
    height = image.getHeight();
    Pick $Pixel_{x,y}$ where $0 \geq x \leq$ width and $0 \geq y \leq$ height;
  Pcolour = image.getRGB(x,y);
  for(int x = 0; x < image.getWidth(); x++)
  for(int y = 0; y < image.getHeight(); y++){
  current_Pixelcolour = image.getRGB(x,y);
    if current_pixelcolour is within range
            Project the original colour;
            else
            Change colour to white;
    }
Save image;
End
```

Listing 2: Algorithm for colour range highlight.

## 4 Results and discussions

Figure 2(a-d) shows some of the visual outputs of group 1 for both NI (a & b) and CGI (c & d). From figure 2 (a & b), it was observed that in NI, a colour that appears to be the same visually is actually represented by a wide range of RGB codes. This could be largely due to the demosaicking process that NI undergo while being produced or the lighting conditions when the image was captured. Therefore picking a random pixel colour and projecting all pixels with exactly the same colour code yielded a scanty set of pixels visually. However, for CGI, the visual results are considerably different. The CGI visual results show that a higher number of pixels are projected for group 1. This exact colour projection sometimes corresponded with a 'shape' in the CGI as can be seen in figure 2 (c & d). This could be because most CGI are a combination of various shapes, where each shape is "filled" with the same colour and then the colour of some areas "blended".

For the colour range highlight, although eight groups were initially proposed, it was observed that beyond group 4 the number of projected pixels remained almost constant for both CGI and NI. This can be viewed from the projected pixel count result displayed in listing 3 (for a natural image) and listing 4 (for a computer generated image). The listing includes the file chosen, its resolution, the pixel chosen $(A_{x,y})$, the pixel colour $RGB(A_{x,y})$ and finally the various counts of projected pixels by range. Listing 3 showed that 37 pixels were projected for group 1; 98 pixels for group 2, and so on. This result showed that for NI, there is usually a gradual increase in the number of projected pixels from group 1 to group 4. Listing 4 however showed that 3242 pixels were projected for group 1; 3488 pixels for groups 2 and beyond. This showed that computer generated images projected almost a constant number pixels.

```
        File Chosen is C:\...\Natural Images\4Egg.jpg
        Image Width: 1918Image Height: 1077
        Chosen Pixel is: 833,392
        Java RGB code is : -1920995
        the real RGB values are: Alpha: 255, Red: 226, Green: 176, Blue: 29
        For range 0       : ProjectedCount is 37
        For range 10      : ProjectedCount is 98
        For range 20      : ProjectedCount is 246
        For range 30      : ProjectedCount is 267
        For range 40      : ProjectedCount is 267
        For range 50      : ProjectedCount is 267
        For range 60      : ProjectedCount is 267
        For range 70      : ProjectedCount is 267
```

Listing 3: Projected pixel count for a selected natural image.

```
File Chosen is C:\...\CGI\ tamar8.jpg
        Image Width: 564  Image Height: 942
        Chosen Pixel is: 114,194
        Java RGB code is : -13171452
        the real RGB values are: Alpha: 255, Red: 55, Green: 5, Blue: 4
        For range 0       : ProjectedCount is 3242
        For range 10      : ProjectedCount is 3488
        For range 20      : ProjectedCount is 3488
        For range 30      : ProjectedCount is 3488
        For range 40      : ProjectedCount is 3488
        For range 50      : ProjectedCount is 3488
        For range 60      : ProjectedCount is 3488
        For range 70      : ProjectedCount is 3488
```

Listing 4: Projected pixel count for a selected computer generated image.

The projected pixel counts for groups 1 to 4 were saved, processed and analysed. Using equations 1-4, the average percentages, $P_1$, $P_2$, $P_3$, $P_4$ of projected pixels were calculated for groups 1, 2, 3 and 4 respectively.

$$P_1 = \frac{\sum C_{r0}}{\sum C_{r\pm30}} \times \frac{100}{1} \qquad \text{Equation (1)}$$

$$P_2 = \frac{\sum C_{r\pm10}}{\sum C_{r\pm30}} \times \frac{100}{1} \qquad \text{Equation (2)}$$

$$P_3 = \frac{\sum C_{r\pm20}}{\sum C_{r\pm30}} \times \frac{100}{1} \qquad \text{Equation (3)}$$

$$P_4 = \frac{\sum C_{r\pm30}}{\sum C_{r\pm30}} \times \frac{100}{1} \qquad \text{Equation (4)}$$

Where:

$C_{r0}$ = count of projected pixels for group 1

$C_{r\pm10}$ = count of projected pixels for group 2

$C_{r\pm20}$ = count of projected pixels for group 3

$C_{r\pm30}$ = count of projected pixels for group 4

These equations were then optimised to give a generalized equation 5

$$P_i = CCG_i \times CCG_j^{-1} \times K \qquad \text{Equation (5)}$$

Where

$P_i$ is the percentage of projected pixel for a group $i$

$i = 1, 2, 3, 4$ ; $j = 4$; $CCG$ is count of projected pixels for groups $i$ or $j$ and $K$ is a constant.

The summary of the analysed data is presented in table 1. From table 1 it can be observed that the average

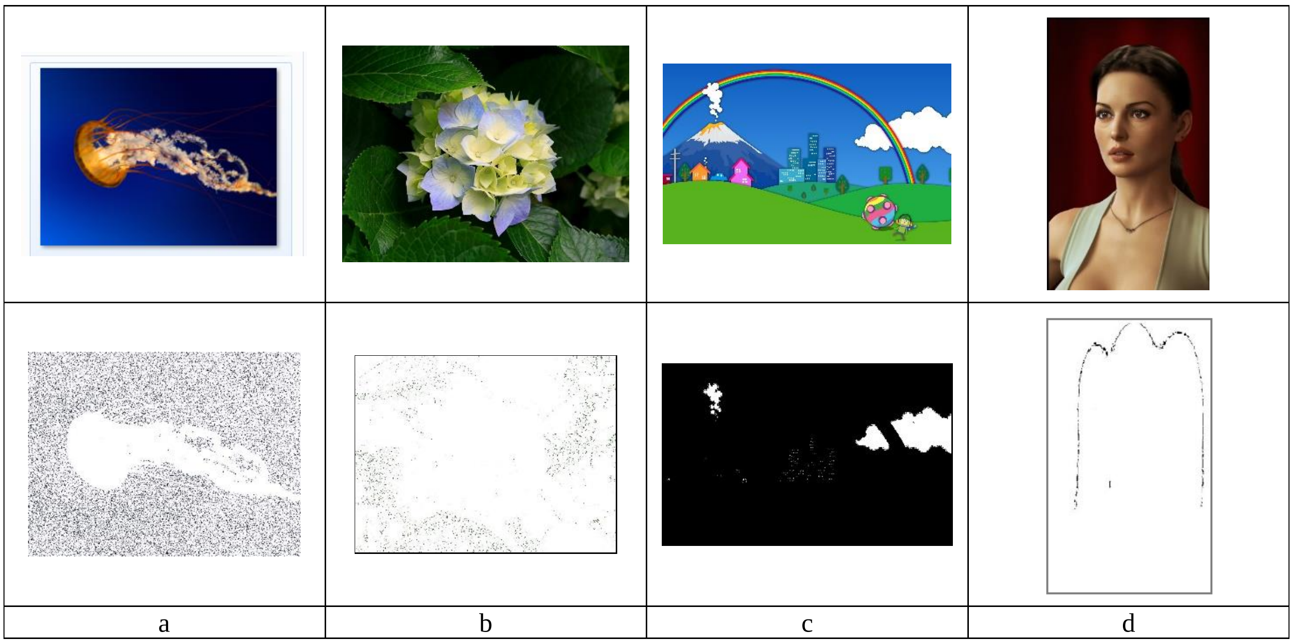

Figure 2: Results of projected exact colour areas in some images.

percentages of projected pixels for natural images were 31.07%, 82.64%, 90.75% and 95.00% for groups 1, 2, 3 and 4, respectively, while CGI projected an average of 69.79%, 92.87%, 96.87% and 98.60%, of any colour code for groups 1, 2, 3 and 4, respectively.

| | NI % projected | CGI % projected |
|---|---|---|
| $P_1$ | 31.07 | 69.79 |
| $P_2$ | 82.64 | 92.87 |
| $P_3$ | 90.75 | 96.87 |
| $P_4$ | 95.00 | 98.60 |

Table 1: Average % pixel colour projection.

This shows that natural images contain a wide range of RGB colour codes for a particular colour that has similar visual colour presentation [20].

For each image, the value of $P_1, P_2, P_3, P_4$ were further analysed in order to distinguish between NI and CGI. The analysis showed that an image is classified as CGI if:

$$AND(P_2 - P_1 \leq 60;\ P_3 - P_2 \leq 30;\ P_4 - P_3 \leq 15)$$

While an image is classified as NI if:

$$NAND(P_2 - P_1 \leq 25;\ P_3 - P_2 \leq 12;\ P_4 - P_3 \leq 6)$$

Using the above results we achieved the following classification percentages

| | CGI | NI |
|---|---|---|
| True Positives | 81.6% | 87.0% |
| False negatives | 18.4% | 13.0% |

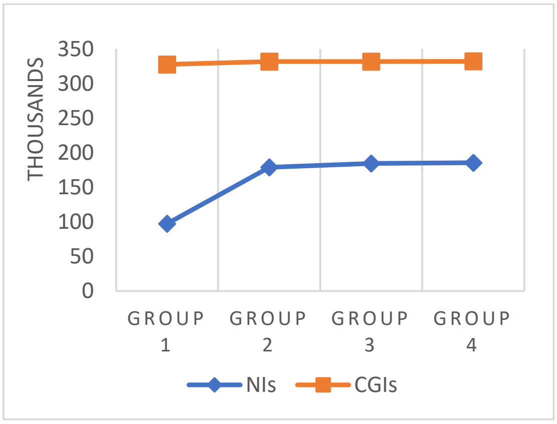

Figure 3: Average number of projected pixels.

Figure 3 shows a graph of the total number of projected pixels for all the computer generated images and natural images in the sample size. This figure shows that irrespective of the random colour chosen, computer generated images projected almost a "constant" number of pixels across the four groups, this can be seen in figure 3 where the red line for computer generated images is almost a horizontal straight line. The pattern of the blue line for the natural images shows a sharp increase in the number pixels emphasizes from group 1 to group 2 and then a gradual increase from group 2 to group 4. The figure also showed that CGI projected a greater percentage of their total pixels than natural images, despite the fact that most natural images had greater number of pixels than the computer generated images. Consequently, the disparity in percentage emphasised can be used to differentiate computer generated images from natural images.

## 5 Conclusion

In this research work, the RGB colour features of some selected pixels in both natural and computer generated digital images were extracted, grouped and analysed. The analysis revealed that there is a disparity in the percentage selected/emphasized for the two groups of images. Consequently, this disparity in percentage of colours projected, within range 0 to 40 from a point of reference, can be used as a quick method to differentiate computer generated images from natural images.

## References

[1] Oracle, "The Java tutorials," 2 Febuary 2012. [Online]. Available: http://www.oracle.com/java-se-7-tutorial-2012-02-28-1536013.html. [Accessed 14 June 2013].

[2] M. K. Johnson, K. Dale, S. Avidan, H. Pfister, W. T. Freeman and W. Matusik, (2011) "CG2Real: Improving the Realism of Computer Generated Images using a Large Collection of Photographs," IEEE Transactions on Visualization and Computer Graphics, vol. XVII, no. 9, pp. 1273 - 1285. https://doi.org/10.1109/tvcg.2010.233

[3] T.-t. Ng, S.-f. Chang, C.-y. Lin and Q. Sun, (2006) "Passive-blind Image Forensics," in In Multimedia Security Technologies for Digital Rights, Elsevier, pp. 383- 412. https://doi.org/10.1016/b978-012369476-8/50017-8

[4] G. K. Birajdar and V. H. Mankar, (2013). "Digital image forgery detection using passive techniques: A survey," Digital Investigation, vol. 10, no. 3, pp. 226-245. https://doi.org/10.1016/j.diin.2013.04.007

[5] G. K. Birajdar and V. H. Mankar, (2017). "Computer Graphic and Photographic Image Classification using Local Image Descriptors," Defence Science Journal, vol. 67, no. 6, pp. 654-663. https://doi.org/10.14429/dsj.67.10079

[6] F. Peng, J. Liu and M. Long, (2012). "Identification of Natural Images and Computer Generated Graphics Based on Hybrid Features," International Journal of Digital Crime and Forensics, vol. IV, no. 1, pp. 1-16. https://doi.org/10.4018/jdcf.2012010101

[7] A. Swaminathan, M. Wu and K. J. R. Liu, (2006). "Component forensics of digital cameras: A non-intrusive approach," in 2006 40th Annual Conference on Information Sciences and Systems. https://doi.org/10.1109/ciss.2006.286646

[8] M. Chandra, S. Pandey and R. Chaudhary, (2010). "Digital watermarking technique for protecting digital images," in 3rd IEEE International

Conference on Computer Science and Information Technology (ICCSIT).
https://doi.org/10.1109/iccsit.2010.5565177

[9] A. C. Popescu and H. Farid, (2005). "Exposing digital forgeries in color filter array interpolated images," IEEE Transactions on Signal Processing, vol. 53, pp. 3948-3959.
https://doi.org/10.1109/tsp.2005.855406

[10] Wang W., Dong J., Tan T. (2009) A Survey of Passive Image Tampering Detection. In: Ho A.T.S., Shi Y.Q., Kim H.J., Barni M. (eds) Digital Watermarking. IWDW 2009. Lecture Notes in Computer Science, vol 5703. Springer, Berlin, Heidelberg
https://doi.org/10.1007/978-3-642-03688-0_27

[11] S. D. Mahalakshmia, K. Vijayalakshmib and S. Priyadharsinia, (2012). "Digital image forgery detection and estimation by exploring basic image manipulations," Digital Investigation, pp. 215-225.
https://doi.org/10.1016/j.diin.2011.06.004

[12] S. Lyu and H. Farid, (2005). "How realistic is photorealistic?," IEEE Transactions on Signal Processing, vol. 53, no. 2, pp. 845-850.
https://doi.org/10.1109/tsp.2004.839896

[13] J. Lukas, J. Fridrich and M. Goljan, (2005). "Determining digital image origin using sensor imperfections," in SPIE Electronic Imaging, Image and Video Communication and Processing, San Jose, California.
https://doi.org/10.1117/12.587105

[14] H. Farid and S. Lyu, (2003). "Higher-order wavelet statistics and their application to digital forensics," in Computer Vision and Pattern Recognition.
https://doi.org/10.1109/cvprw.2003.10093

[15] T.-T. Ng, S.-F. Chang, J. Hsu, L. Xie and M.-P. Tsui, (2005). "Physics-motivated features for distinguishing photographic images and computer graphics," in ACM Multimedia, Singapore.
https://doi.org/10.1145/1101149.1101192

[16] S. Dehnie, T. Sencar and N. Memon,. (2006). "Digital Image Forensics for Identifying Computer Generated and Digital Camera Images," in IEEE International Conference on image processing.
https://doi.org/10.1109/icip.2006.312849

[17] A. E. Dirik, S. Bayram, H. T. Sencar and N. Memon, (2007). "New features to identify computer generated images," in IEEE International Conference on Image Processing 4.
https://doi.org/10.1109/icip.2007.4380047

[18] X. Kang, E. Zhang, Y. Chen and Y. Wei, (2011). "Forensic discrimination of computer generated images and photographs using spectral correlations in wavelet domain," Energy Procedia, vol. 13, no. 311, pp. 2174-2182.
https://doi.org/10.1016/S1876-6102(14)00454-8

[19] F. Peng, Y. Zhu and M. Long, (2015). "Identification of Natural Images and Computer Generated Graphics using Multi-fractal Differences of PRNU," in ICA3PP 2015: Part II of the 15th International Conference on Algorithms and Architectures for Parallel Processing.
https://doi.org/10.1007/978-3-319-27122-4_15

[20] N. C. Woods and C. A. B. Robert, (2017) "A Model for Creating Exact Colour Spectrum for Image Forensic," University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR), vol. Volume 1, no. 1, pp. 1-6.